## REMARKS

The examiner rejected claims 1 and 3-21 under U.S.C. § 112, second paragraph, as being Indefinite for falling to particularly point out and distinctly claim the subject matter which the applicants regard as the invention. Specifically, the examiner contends that

> **"[w]ith respect to independent claims [1, 7, and 15], it is not clear whether the thread is for processing data blocks from the MAC device or for moving the incoming blocks from the MAC device to memory locations, or both. If it is both, it is not clear which one is performed first [Office Action, Page 2].**

Applicants contend that independent claims 1, 7, and 15 clearly call for scheduling a first thread … to process a first incoming block of data within a network packet received at a port of a media access control device to move the first incoming block of data to a first location in a memory … and scheduling a second thread … to process a second incoming block of data … .. Support for this can be found in the Applicants' specification. "The receive scheduler 92 assigns a ready port to an available receive thread, and causes transfer of packet data into the network processor. The receive threads then analyze and store sections of incoming packet data (MPKTs)."[1] "Thread X 303…receives control information and saves state, …and then continues to process its network data."[2] "For every receive request transaction the port bus interface transfers a section of the packet from a MAC device port to a receive FIFO in the network processor…After saving the updated address, the thread copies the data from the RFIFO to DRAM…The receive thread writes a thread done register to indicate it is available for re-assignment."[3]

Therefore, Claims 1 and 3-21 particularly point out and distinctly claim the subject matter which the applicants regard as the invention. Applicants request that the rejection of claims 1 and 3-21 under U.S.C. § 112, second paragraph, be withdrawn.

---

[1] Specification, Page 23, Lines 7-9.
[2] Id., Page 23, Lines 16-18.
[3] Id., Page 25, Lines 6-17.

103 rejections

The examiner rejected claims 1 and 3-21 under U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,212,542 ("Kahle") in view of U.S. Patent No. 6,604,125 ("Belkin"). The examiner contends that, with respect to claims 1, 7, and 15:

> **Kahle teaches…a method of processing network data in a processor (multiscalar processor) having multiple programmable multi-threaded engines (PEs, 132-138 Figure 4) integrated within the processor, the method comprising: scheduling (thread scheduler. Fig. 4) a first thread provided by the multiple programmable multi-threaded engines integrated within the processor to process a first incoming block of data within a network packet received at port of a media access control device ; and scheduling a second thread provided by the multiple programmable multithreaded engines integrated within the processor to process a second incoming block of data within the network packet prior to the first thread completing processing of the first incoming block of data.**
>
> **The only difference is that Kahle does not make clear whether his processor is processing in a network environment. Using a processor having multiple thread engines to process network data is well known in the art. Belkin teaches using a processor having multiple thread engines to process network data. From the teaching of Belkin, it would have been obvious to a person of ordinary skill in the art to use the multiscalar processor of Kahle in a network environment so that packets can be processed by the multiple engines (PEs)…**
>
> **Further with respect to the amendment, it is well known that incoming data blocks from a network are inherently stored in memory locations, buffer in an I/O device or memory of a processor, before they are being accessed and processed by the processor…**
>
> **With respect to the argument directed to the limitation "media access control device, note that in Belkind [sic] data is received from a network which is a media. The I/O device (inherent) in Belkin for receiving data from the network is a media access control device.**

Applicants contend that Kahle does not teach "multiple programmable multi-threaded engines integrated within [a] processor", as required by claims 1, 7, and 15. Kahle teaches the following about the processing elements (PEs) that the examiner likens to "programmable multi-threaded engines":

> **Referring to PEs 132-138 [in Fig. 4], the central component of each of PEs 132-138 is an execution core 158 that executes instructions contained within an assigned thread. In a preferred embodiment, execution core 158**

**contains superscalar circuitry that supports intra-thread branch speculation and includes multiple execution units capable of executing multiple ISA instructions out-of-order during each cycle. However, based upon design and cost considerations, execution core 158 of PEs 132-138 can alternatively employ any one of a number of diverse hardware architectures. For example, execution core 158 may comprise a single execution resource that executes ISA instructions sequentially. Regardless of which hardware architecture is utilized to implement execution core 158, each execution core 158 includes an instruction sequencer that fetches and dispatches instructions and at least one execution resource that executes instructions.**

**Local storage is provided to each execution core 158 by an associated instruction cache 150, data cache 156, and GPR cache 154, which respectively store the ISA instructions, memory data values, and data and condition register values required by the associated execution core 158 during execution. Each execution core 158 is also coupled to CAM 160 that stores the extension list associated with the thread executing within the associated execution core 158. Extension instructions in the extension list are dynamically inserted into the thread executed by the associated execution core 158 in accordance with the method described below with respect to FIG. 8.**

**Each of PEs 132-138 further includes communication and synchronization logic 152, which is coupled to both GPR cache 154 and data cache 156. Communication and synchronization logic 152 maintains register and memory data coherency (i.e., the availability of data to the associated PE) through inter-PE and PE-L2 communication across local communication and synchronization mechanism 170, which, in order to reduce latency, preferably includes four concurrent address busses for register communication and at least one address bus for memory communication. Communication across local communication and synchronization mechanism 170 is performed under the arbitrating control of arbitration logic 172. [Kahle, Column 11, Line 35 – Column 12, line 9]**

The PEs of Kahle do possess an instruction cache that stores ISA instructions, and an execution core that executes instructions. Moreover, nowhere does Kahle teach that any of these PEs are programmable as required by claims 1, 7, and 15.

Applicants contend that Belkin does not cure the infirmities of Kahle. Nowhere does Belkin disclose or suggest "multiple programmable multi-threaded engines integrated within [a] processor". Belkin teaches the following:

**In servicing the requests, the server 106 implements multi-threading. That is, within the server 106, there are multiple threads of execution, with each thread capable of independent execution. Because each thread is capable of executing independently, multiple threads can service multiple requests concurrently. In addition, the server 106 implements multiple thread pools.**

> **Each thread pool comprises one or more threads, and each thread pool has associated with it a set of characteristics. In one embodiment, the characteristics of each thread pool are set such that they are customized for one or more types of service. Each of the threads in the server 106 belongs within one of the thread pools. [Belkin, Column 4, Lines 53-65]**

The system of Belkin relies on multiple thread pools within a server. Assuming *arguendo*, that each thread pool could be considered to be a multi-threaded engine, Belkin would still not satisfy the requirements of Applicants' claims 1, 7, and 15 because such engines are not integrated within a processor.

Belkin makes explicit mention of "engines" within a service or set of services in a web server within the multithreaded environment:

> **In server 106, the request processing mechanism 110 is the component primarily responsible for receiving incoming requests, determining which service or services 112 need to be invoked to service the requests, and then assigning threads from the appropriate thread pools to be used by the services 112 to service the requests. In assigning the threads, the request processing mechanism 110 uses the information in the set of thread pool information 114 to determine the proper thread pools from which the threads should be taken.**
> **Depending upon the request, various services 112 within the server 106 may be invoked in order to service the request. For example, if the request is a simple request for an HTML page, then the request processing mechanism 110 forwards the request on to HTML engine 122 for further servicing. On the other hand, if the request is for a common gateway interface (CGI) program, then the request processing mechanism 110 forwards the request on to the CGI engine 124 for servicing. In response, the CGI engine 124 invokes one or more CGI applications 130. If the request is for a JAVA type service, then the request processing mechanism 110 forwards the request on to the JAVA services engine 126. In turn, the JAVA services engine 126 invokes one or more JAVA type applications 132 (e.g. servelets, server pages, JAVA programs, etc.). To run JAVA type applications, a JAVA virtual machine (JVM) 116 is needed. Hence, a JVM 116 may be incorporated into the server 106. The request may also request other types of services, such as certain legacy code 120. If that is the case, then the request processing mechanism 110 invokes the legacy code 120 to further service the request. The services 112 mentioned thus far are just some examples of the types of services that may be provided by the server 106. Other services may also be provided within the spirit of the invention. [Belkin, Column 4, Lines 28-63]**

However these engines are not programmable and are not integrated within a multithreaded processor, as required by Applicants' claims 1, 7, and 15. Therefore, the purported combination of Kahle and Belkin neither describes nor suggests "multiple programmable multi-threaded engines integrated within [a] processor", as required by Applicants' claims 1, 7, and 15.

Claims 3-6 and 18-20 depend from claim 1, claims 8-14 and 21 depend from claim 7, and claim 16 and 17 depend from claim 15. Therefore, claims 1 and 3-21 are patentable over the combination of Kahle and Belkin.

The examiner rejected claims 1 and 3-21 under U.S.C. § 103(a) as being unpatentable over Belkin in view of U.S. Patent No. 6,338,078 ("Chang"). The examiner contends that, with respect to claims 1, 7, and 15:

> **[Belkin teaches that] a method of processing network data in a processor (web server 106, Fig. 1) having multiple programmable multi-threaded engines (engines 120-126, Figure 1) integrated within the processor, the method comprising: scheduling (assign scheduler. Fig. 4, column 4/line 66 to column 5/line18) a first thread provided by the multiple programmable multi-threaded engines integrated within the processor to process a first incoming block of data within a network packet received at port of a media access control device ; and scheduling a second thread provided by the multiple programmable multithreaded engines integrated within the processor to process a second incoming block of data within the network packet prior to the first thread completing processing of the first incoming block of data.**
>
> **The only difference is that Belkin does not specify that his network packet is received at a port of a device named media access device. Media access control device is well known in network communication art. Chang teaches media access control device in his network system (column 6/line 35). From the teaching of Chang, it would have been obvious to a person of ordinary skill in the art to use a MAC device in Belkin to access media.**

Belkin, as mentioned above, neither discloses nor suggests "multiple programmable multi-threaded engines integrated within [a] processor". Nor does Belkin disclose or suggest "…scheduling a first thread provided by any of the multiple programmable multi-threaded engines integrated within the processor to process a first incoming block of data within a network packet received at a port of a media access control device …", as required by claim 1. Rather, Belkin states that

> **The thread pool configuration table 200 is just one of the sets of information contained within the thread pool information 114. Another set of information is the association table 300 shown in FIG. 3. It is in this table 300 that the user specifies the association between particular thread pools and other components in the server 106. [Belkin, Column 7, Lines 36-41]**
>
> **…the association table 300a comprises a column 302 for the service type (e.g. JAVA, CGI, HTML, etc.), a column 304 for the pool ID of a particular associated thread pool, and a column for the pointer 306 to the thread pool. [Id., Column 7, Lines 49-53]**

As understood, thread scheduler of Belkin assigns specific threads for specific services. That is, the scheduler does not assign any of the available threads to process a given block of data, as required by the Applicants' claims 1, 7, and 15.

Applicants contend that Chang does not cure the infirmities of Belkin. Chang states that

> **Packets received on the network are distributed to N high priority threads, wherein N is the number of CPUs on the system. N queues are provided to which the incoming packets are distributed. When one of the queues is started, one of the threads is scheduled to process packets on this queue. When all of the packets on the queue are processed, the thread becomes dormant. Packets are distributed to one of the N queues by using a hashing function…The hashing mechanism ensures that the sequence of packets within a given communication session will be preserved. [Chang, Column 2, Line 57 – Column 3, Line 2]**
>
> **…in accordance with the invention depicted in FIG. 3, this IP queue is concurrently processed by multiple threads with one thread per CPU, and one queue per thread. In this manner the path length is shortened with more CPUs running parts of a packet's code simultaneously in a multithread fashion, thereby effecting the increased throughput. [Id., Column 5, Lines 20-25]**

Chang therefore requires that each CPU (engine) have exactly one thread for increased throughput. Chang, however, neither discloses nor suggests "multiple programmable multi-threaded engines integrated within [a] processor" as required by Applicants' claims 1, 7, and 15.[4] Therefore, the purported combination of Belkin and Chang neither discloses nor suggests

[4] Applicants note that claim 4 of Chang implies that the number of threads may take on a different value than the number of CPUs (engines). Nevertheless, this claim is never taught in Chang's specification. Moreover, Chang makes it absolutely clear in the passages quoted that one thread per CPU is an important part of the effectiveness of Chang's invention.

"multiple programmable multi-threaded engines integrated within [a] processor", as required by Applicants' claims 1, 7, and 15.

Claims 3-6 and 18-20 depend from claim 1, claims 8-14 and 21 depend from claim 7, and claim 16 and 17 depend from claim 15. Therefore, claims 1 and 3-21 are patentable over the combination of Belkin and Chang.

It is believed that all the rejections and/or objections raised by the examiner have been addressed.

In view of the foregoing, applicant respectfully submits that the application is in condition for allowance and such action is respectfully requested at the examiner's earliest convenience.

Canceled claims, if any, have been canceled without prejudice or disclaimer.

Any circumstance in which the applicant has (a) addressed certain comments of the examiner does not mean that the applicant concedes other comments of the examiner, (b) made arguments for the patentability of some claims does not mean that there are not other good reasons for patentability of those claims and other claims, or (c) amended or canceled a claim does not mean that the applicant concedes any of the examiner's positions with respect to that claim or other claims.

Please apply any other charges or credits to deposit account 06-1050, referencing attorney docket no. 10559-137002.

Respectfully submitted,

Date:January 29, 2008

/Denis G. Maloney/
Denis G. Maloney
Reg. No. 29,670

Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110
Telephone: (617) 542-5070
Facsimile: (617) 542-8906